UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/678,846	10/03/2003	Gordon Bowman	GLH 08-896305	2420

27667 7590 05/23/2008

HAYES SOLOWAY P.C.
3450 E. SUNRISE DRIVE, SUITE 140
TUCSON, AZ 85718

EXAMINER	
KISS, ERIC B	

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
05/23/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/678,846	BOWMAN ET AL.	
	Examiner	Art Unit	
	Eric B. Kiss	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 February 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-26* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-26* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. The reply filed February 8, 2008, has been received and entered. Claims 1-26 are pending.

Response to Arguments

2. Applicant's arguments filed February 8, 2008, have been fully considered but they are not persuasive. Specifically, the examiner asserts that Claussen et al. discloses predefined document object model behavior elements in that the tag libraries defining appropriate tag handlers are created in advance and registered through an XML taglib tag and specified through a URI (see, e.g., col. 5, line 45, through col. 6, line 48 (tag libraries are registered listing recognized tags and directives on how to load the appropriate tag handlers during runtime processing)).

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 1-11 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 1, 10, and 11 each recite a processor for executing instructions, the executed instructions "implementing" recited components of the claims. These claims are indefinite because it is unclear what concrete steps are involved in such "implementation." Specifically, it is unclear whether implementing the components is intended to merely cause the components to exist (i.e., become stored in memory) or whether the execution of instructions is intended to put existing components to some

functional use (e.g., whether the implemented components are themselves the executable instructions). With regard to the scripts recited in the claims, it is unclear whether "implementing" these scripts is intended to mean creating the scripts or actually executing the scripts (the claims appear to recite executing instructions to implement scripts and not necessarily implementing scripts.

Claim Rejections - 35 USC § 101

5. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

6. Claims 1-11 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Descriptive material can be characterized as either "functional descriptive material" or "nonfunctional descriptive material." In this context, "functional descriptive material" consists of data structures and computer programs which impart functionality when employed as a computer component. (The definition of "data structure" is "a physical or logical relationship among data elements, designed to support specific data manipulation functions." The New IEEE Standard Dictionary of Electrical and Electronics Terms 308 (5th ed. 1993).) "Nonfunctional descriptive material" includes but is not limited to music, literary works and a compilation or mere arrangement of data. Both types of "descriptive material" are nonstatutory when claimed as descriptive material *per se*. *In re Warmerdam*, 33 F.3d 1354, 1361, 31 USPQ2d 1754, 1760 (claim to a data structure per se held nonstatutory).

Data structures not claimed as embodied in computer-readable media are descriptive material *per se* and are not statutory because they are not capable of causing functional change in the computer. *See, e.g., In re Warmerdam*, 33 F.3d 1354, 1361, 31 USPQ2d 1754, 1760 (claim to a data structure per se held nonstatutory). Such claimed data structures do not define any structural and functional interrelationships between the data structure and other claimed aspects of the invention which permit the data structure's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a data structure defines structural and functional interrelationships between the data structure and the computer software and hardware components which permit the data structure's functionality to be realized, and is thus statutory.

Similarly, computer programs claimed as computer listings per se, *i.e.*, the descriptions or expressions of the programs, are not physical "things." They are neither computer components nor statutory processes, as they are not "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer which permit the computer program's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a computer program is a computer element which defines structural and functional interrelationships between the computer program and the rest of the computer which permit the computer program's functionality to be realized, and is thus statutory. *See In re Lowry*, 32 F.3d 1579, 1583-84, 32 USPQ2d 1031, 1035.

Claims 1-11 recite systems comprising a series of elements including hardware elements that might be capable of defining a structural and functional interrelationship so

as to achieve a practical application of the recited functional descriptive material. However, as noted above (see the rejection under 35 U.S.C. § 112), the functional interrelationship between the recited elements is unclear from the claims, and thus, the functional result of the claims is uncertain. Because of this uncertainty, claims 1-11 cannot be said to clearly recite statutory subject matter.

7. To expedite a complete examination of the instant application, the claims rejected under 35 U.S.C. §101 (non-statutory) above are further rejected as set forth below in anticipation of Applicant amending these claims to place them within the four statutory categories of invention.

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

9. Claims 1-9 and 11-26 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,981,211 (Claussen et al.).

Regarding claim 1, *Claussen et al.* discloses a system for manipulating a document object model, the system comprising:

a memory for storing instructions (see, e.g., FIG. 1);

a processor for executing the instructions stored in memory (see, e.g., FIG. 1), the executed instructions implementing:

a collection of predefined document object model behavior elements (see, e.g., col. 5, line 45, through col. 6, line 48 (tag libraries are registered listing recognized tags and directives on how to load the appropriate tag handlers during runtime processing); col. 7, line 15-16), each behavior element comprising:

a namespace (see, e.g., col. 7, lines 20-22);

an event attribute for associating the behavior element to an event (see, e.g., col. 5, lines 45-67 (tag libraries are registered listing recognized tags and directives on how to load the appropriate tag handlers during runtime processing)); and

other attributes for describing features of the behavior element (see, e.g., col. 8, lines 26-36); and

a collection of scripts for performing actions associated with the set of behavior elements, each script associated with a behavior element (see, e.g., col. 5, line 45, through col. 6, line 48 (taglibs defining tag handlers)).

Regarding claims 2-4, *Claussen et al.* further discloses the behavior element is associated with (parent/child of) an extensible markup language element (see, e.g., col. 19, lines 42-51 (describing processing of parent/child nodes); col. 9, lines 2-27 (processing the DOM tree as XML)).

Regarding claim 5, *Claussen et al.* further discloses the actions comprise behavioral mutations of an output of extensible markup language elements (see, e.g., col. 8, line 51, through col. 9, line 27).

Regarding claim 6, *Claussen et al.*further discloses an initialization function for directing the processing of one or more behavior elements in a document object model, the initialization function having instructions for traversing each node in the document object model and for searching and calling functions associated with behavior elements having names following the predetermined naming convention ((see, e.g., col. 5, line 45, through col. 6, line 48 (tag libraries are registered listing recognized tags and directives on how to load the appropriate tag handlers for custom tags in a DOM during runtime processing)).

Regarding claim 7, *Claussen et al.* further discloses:

a collection of behavior attributes for adding to existing regular extensible markup language elements in a document object model, the behavior attributes following the predetermined naming convention (see, e.g., col. 5, line 45, through col. 6, line 48 (taglibs defining tag handlers for custom tags in a DOM)); and

a collection of scripts for performing actions associated with the collection of behavior attributes, each script associated with a behavior attribute (see, e.g., col. 5, line 45, through col. 6, line 48 (taglibs defining tag handlers)).

Regarding claim 8, *Claussen et al.* further discloses the initialization function contains instructions for traversing each node in the document object model and for searching and calling functions associated with behavior elements and behavior attributes having names following the predetermined naming convention (see, e.g., col. 5, line 45, through col. 6, line 48 (tag libraries are registered listing recognized tags and directives on how to load the appropriate tag handlers during runtime processing)).

Regarding claim 9, *Claussen et al.* further discloses the collection of behavior elements comprises a markup language (see, e.g., col. 5, line 45, through col. 6, line 3).

Regarding claim 11, *Claussen et al.* discloses a system for manipulating a document object model (see, e.g., col. 5, lines 3-6), the system comprising:

a memory for storing instructions (see, e.g., FIG. 1);

a processor for executing the instructions stored in memory (see, e.g., FIG. 1), the executed instructions implementing:

a collection of scripts for performing actions associated with markup behavior elements, each script associated with a predefined behavior element (see, e.g., col. 5, line 45, through col. 6, line 48 (taglibs defining tag handlers)); and

an initialization function for directing the processing of one or more behavior elements in a document object model (see, e.g., col. 5, line 45, through col. 6, line 48 (the taglib specifies, among other things, directives on how to load the appropriate tag handlers)).

Regarding claim 12, *Claussen et al.* discloses a method of manipulating a document object model (see, e.g., col. 5, lines 3-6), the method comprising the steps of:

searching for a predefined designated element in a document object model (see, e.g., col. 5, line 45, through col. 6, line 48 (taglibs defining tag handlers)); and

calling a script associated with the designated element (see, e.g., col. 6, lines 36-48; col. 7, lines 11-12).

Regarding claim 13, *Claussen et al.* further discloses the step of searching includes the steps of:

traversing each node in the document object model (see, e.g., col. 7, lines 37-51); and

determining whether an element has a name which follows a designated naming convention (see, e.g., col. 7, lines 37-51).

Regarding claim 14, *Claussen et al.* further discloses the step of calling a script includes the steps of:

dynamically generating a function name associated with the designated element (see, e.g., col. 5, lines 10-12; col. 7, lines 2-12);

passing an object associated with the designated element as a parameter of the generated function (see, e.g., col. 8, lines 26-36);

retrieving the attributes of the object (see, e.g., col. 8, lines 26-36); and

performing a function stored in memory having the generated function name (see, e.g., col. 7, lines 11-12).

Regarding claim 15, *Claussen et al.* further discloses the step of dynamically generating includes the steps of:

determining if the name of the designated element contains a designated prefix (see, e.g., col. 7, lines 37-51);

generating a function name comprising of the name of the designated element (see, e.g., see, e.g., col. 7, lines 2-12);

assigning an object associated with the designated element as the parameter of the function (see, e.g., col. 8, lines 26-36); and

assigning predetermined instructions of the designated element as steps for the function to perform (see, e.g., col. 7, lines 2-12; col. 8, lines 2-48).

Regarding claim 16, *Claussen et al.* further discloses the step of calling a script includes the steps of:

determining which script in a collection of scripts is associated with the designated element (see, e.g., col. 5, lines 45-67 (tag libraries are registered listing recognized tags and directives on how to load the appropriate tag handlers during runtime processing)); and

calling the script (see, e.g., col. 6, lines 36-48).

Regarding claim 17, *Claussen et al.* further discloses:

searching for a designated attribute in an element in a document object model (see, e.g., col. 5, line 45, through col. 6, line 48 (taglibs defining tag handlers)); and

calling a script associated with the designated attribute (see, e.g., col. 6, lines 36-48; col. 7, lines 11-12).

Regarding claim 18, *Claussen et al.* further discloses the step of searching for a designated attribute comprises the steps of:

searching attributes of an element in a document object model (see, e.g., col. 5, line 45, through col. 6, line 48 (taglibs defining tag handlers));

determining whether an element attribute has a name which follows a designated naming convention (see, e.g., col. 7, lines 37-51).

Regarding claim 19, *Claussen et al.* further discloses the step of calling a script includes the steps of:

determining if the name of the designated attribute contains a designated prefix (see, e.g., col. 7, lines 37-51);

generating a function name comprising the name of the designated attribute (see, e.g., see, e.g., col. 7, lines 2-12);

assigning an object associated with the designated attribute as the parameter of the function (see, e.g., col. 8, lines 26-36); and

assigning predetermined instructions of the designated attribute as steps for the function to perform (see, e.g., col. 7, lines 2-12; col. 8, lines 2-48).

Regarding claim 20, *Claussen et al.* further discloses the step of calling a script includes the steps of:

dynamically generating a function name associated with the designated element (see, e.g., col. 5, lines 10-12; col. 7, lines 2-12);

passing an object associated with the designated element as a parameter of the generated function (see, e.g., col. 8, lines 26-36);

retrieving the attributes of the object (see, e.g., col. 8, lines 26-36); and

performing a function stored in memory having the generated function name (see, e.g., col. 7, lines 11-12).

Regarding claim 21, *Claussen et al.* further discloses the step of dynamically generating comprises the steps of:

determining if the name of the designated element contains a designated prefix (see, e.g., col. 7, lines 37-51);

generating a function name comprising of the name of the designated element (see, e.g., see, e.g., col. 7, lines 2-12);

assigning an object associated with the designated element as the parameter of the function (see, e.g., col. 8, lines 26-36); and

assigning predetermined instructions of the designated element as steps for the function to perform (see, e.g., col. 7, lines 2-12; col. 8, lines 2-48).

Regarding claim 22, *Claussen et al.* further discloses the step of calling a script includes the steps of:

determining which script in a collection of scripts is associated with the designated element (see, e.g., col. 5, lines 45-67 (tag libraries are registered listing

recognized tags and directives on how to load the appropriate tag handlers during runtime processing)); and

calling the script (see, e.g., col. 6, lines 36-48).

Regarding claim 23, *Claussen et al.* discloses a method of manipulating a document object model, the method comprising the steps of:

adding an event listener to an element having a predefined designated element as a child in the document object model (see, e.g., col. 5, lines 45-67 (tag libraries are registered listing recognized tags and directives on how to load the appropriate tag handlers during runtime processing));

receiving an event which is equal to an event attribute setting in the designated element (col. 6, lines 36-48 (A tag with a corresponding tag handler is processed)); and

calling a script associated with the designated element (see, e.g., col. 6, lines 36-48).

Regarding claim 24, *Claussen et al.* further discloses the step of calling a script includes the steps of:

determining if the name of the designated element contains a designated prefix (see, e.g., col. 7, lines 37-51);

generating a function name comprising of the name of the designated element (see, e.g., see, e.g., col. 7, lines 2-12);

assigning an object associated with the designated element as the parameter of the function name (see, e.g., col. 8, lines 26-36); and

assigning predetermined instructions of the designated element as steps for a function having the function name to perform (see, e.g., col. 7, lines 2-12; col. 8, lines 2-48).

Regarding claim 25, *Claussen et al.* further discloses the step of calling a script includes the steps of:

dynamically generating a function name associated with the designated element (see, e.g., col. 5, lines 10-12; col. 7, lines 2-12);

passing an object associated with the designated element as a parameter of the generated function name (see, e.g., col. 8, lines 26-36);

receiving the attributes of the object (see, e.g., col. 8, lines 26-36); and

performing a function stored in memory having the generated function name (see, e.g., col. 7, lines 11-12).

Regarding claim 26, *Claussen et al.* discloses a method of creating an element for manipulating a document object model, the method comprising the steps of:

categorizing low level actions into behavior groupings (see, e.g., col. 5, line 45, through col. 6, line 48 (taglibs defining tag handlers));

determining common attributes of a behavior grouping (see, e.g., col. 5, line 45, through col. 6, line 48); and

creating a predefined behavior element having the common attributes of the behavior grouping (see, e.g., col. 5, line 45, through col. 6, line 48).

Conclusion

10. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

11. Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Eric B. Kiss whose telephone number is (571) 272-3699. The Examiner can normally be reached on Tue. - Fri., 7:00 am - 4:30 pm. The Examiner can also be reached on alternate Mondays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Tuan Dam, can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Eric B. Kiss/
Eric B. Kiss
Primary Examiner, Art Unit 2192